METHOD AND SYSTEM FOR ANALYZING DATA AND CREATING PREDICTIVE MODELS

COPYRIGHT NOTICE

RELATED APPLICATIONS

[0002] This application claims the benefit of priority under 35 U.S.C. § 119(e) to U.S. provisional application serial no. 60/432,631, filed December 10, 2002, entitled "Method and System for Analyzing Data and Creating Predictive Models," the entirety of which is incorporated by reference herein.

BACKGROUND OF THE INVENTION

Field of the Invention

[0003] The invention relates to the field of statistical data analysis and, more particularly, to a method and system for automatically analyzing data and creating a statistical model for solving a problem or query of interest with minimal human intervention.

Description of the Related Art

[0004] The age of analytics is upon us. Businesses scramble to leverage knowledge culled from customer, enterprise, and third-party data for more effective decision-making and strategic planning. Somewhere, perhaps only in the minds of forward looking executives and managers, resides a corporate Shangri-la, a place where customer and enterprise data fuse seamlessly and transparently with advanced analytical software to provide a stream of clear and reliable business intelligence.

[0005] Unfortunately, those who labor in search of this nirvana often find the path fraught with difficulty. Advanced analytical software typically requires extensive training and/or advanced statistical knowledge and the statistical model building process can be a lengthy and complex one, including such difficulties as data cleansing and preparation, handling missing

values, extracting useful features from large data sets, and translating model outputs into business knowledge. All told, solutions typically require either expensive payroll increases associated with hiring in-house experts or costly consulting engagements.

[0006] Depending on the scope, modeling projects can cost anywhere from $25,000 to $100,000, or more, and take weeks or even months to complete. Some of the tasks involved in building a statistical model based on a large data set include the following steps:

- Identify Target Variable: The analyst must select or, in many cases create, the target variable, which relates to the question that is being addressed. For example, the target variable in a credit screening application might involve whether a loan was repaid or not.
- Data Exploration: The analyst examines the data, computing and analyzing various summary statistics regarding the different variables contained in the data set. This exploratory analysis is undertaken to identify the most useful predictors, spot potential problems that might be caused by outliers or missing values, and determine whether any of the data fields need to be rescaled or transformed.
- Split Data Set: The analyst may randomly split the data into two sets, one of which will be used to build, or train, the model, and the other of which will be used to test the quality of the model once it is built.
- Categorical Variable Preprocessing: Categorical variables are variables such as gender and marital status that possess no natural numerical order. These variables must be identified and handled differently than continuous numerical variables such as age and income.
- Data Cleansing: The data must be cleansed of missing values and outliers. Missing values are, quite literally, missing data. Outliers are "unusual" data that may skew the results of calculations.
- Variable Reduction: Often there is a preference for parsimonious models, and a variety of methods may be employed to attempt to find the most useful predictors within a potentially large set of possible predictors.
- Variable Standardization: After variable reduction, the remaining variables are often re-scaled so that a model based on these variables is not unduly biased by only a few variables.
- Create Model: Determining the coefficients of variables that best describe the correlation between the target variable and the training data.
- Model Selection: Several competing models may be considered.

- Model Validation: Run the model using the test data taken from the original data set. This provides a measure of model accuracy that guards against over-fitting by presenting the model with new cases not used during the model-build stage.

[0007] In conventional methods and systems, the above steps are performed manually and require the expertise not only of one or more trained analysts, but software programmers as well, and significant time to complete the analysis and processing of the data. In many cases, there are too many variables in the data set, which makes it difficult for an untrained user to analyze and process the data. One particularly difficult task, for example, is deciding which variables should be included in creating a statistical model for a given target variable and which variables should be excluded.

[0008] Thus, there is a need for a method and system that can automatically perform such tasks as data cleansing and preparation, handling missing values, identifying and extracting useful features from large data sets, and translating model outputs into business knowledge, with minimal human intervention and without the need for highly trained statisticians to analyze the data. There is a further need for a method and system that can automatically analyze data, and make decisions as to whether the data is, for example, continuous, categorical, highly predictive, or redundant. Such a method and system should also determine for an untrained user which variables in a given data should be used to create a statistical model for solving a particular problem or query of interest. Additionally, there is need for a method and system that can automatically and efficiently build a statistical model based on the selected variables and, thereafter, validate the model.

Brief Summary of the Invention

[0009] The present invention addresses the above and other needs by providing a method and system that automatically performs many or all of the steps described above in order to minimize the difficulty, time and expense associated with current methods of statistical analysis. Thus, the invention provides an automated data modeling and analytical process through which decision-makers at all levels can use advanced analytics to guide their critical business

decisions. In addition to being highly automated and efficient, the method and system of the invention provides a reliable and robust general-purpose data modeling solution.

[0010] In one embodiment, the invention provides easy-to-use software tools that enable business professionals to build and implement powerful predictive models directly from their desktop computers, and apply statistical analytics to a much broader range of business and organizational tasks than previously possible. Since these software tools automate much of the analytical and modeling processes, users with little or no statistical experience can perform statistical analysis more quickly and easily.

[0011] In a further embodiment, the method and system of the invention automatically handles data exploration and preprocessing, which typically takes 50 to 80 percent of an analyst's time during conventional modeling processes.

[0012] In a further embodiment, the method and system of the invention scans an entire data set and performs the following tasks: automatically distinguishes between continuous and categorical variables; automatically handles problem data, such as missing values and outliers; automatically partitions the data into random test and train subsets, to protect against sample bias in the data; automatically examines the relationship between each potential variable to find the most promising predictor variables; automatically uses these variables to build an optimal statistical model for a given target variable; and automatically evaluates the accuracy of the models it creates.

[0013] In another embodiment, variables in a data set are automatically classified as categorical or continuous. In a further embodiment, categorical variables that exhibit high co-linearity with one or more continuous variables are automatically identified and discarded. In a further embodiment, categories within a variable that are not significantly predictive of the target variable are collapsed with adjacent categories so as to reduce the number of categories in the variable and reduce the amount of data that must be considered and processed to create a statistical model.

[0014] In another embodiment, a subset of variables in a data set having a significant predictive value for a given problem or target variable are automatically identified and selected. Thereafter, only those selected variables and the target variable are used to create a statistical model for a problem or query of interest.

[0015] In another embodiment, variables having strong co-linearities or correlation with other variables are automatically identified and eliminated so as to remove statistically redundant variables when building the model. In one embodiment, only non-redundant variables having the highest predictive value (e.g., co-linearity or correlation) with the target variable are retained in order to create the statistical model.

[0016] In a further embodiment, the method and system of the present invention can use univariate analysis, multivariate analysis and/or Principle Component Analytics (PCA) to select variables and build a model. Since multivariate analysis typically requires greater processing time and system resources (e.g., memory) than univariate analysis, in one embodiment, univariate analysis is used to filter out those variables that have weak predictive value or correlation with the target variable.

[0017] In another embodiment, categorical variables contained in the data set are expanded into dummy variables and added to the design matrix along with continuous variables. Since potential co-linearities exist among these variables, whenever there is any pair of variables having a correlation greater than a threshold, the variable that has a weaker correlation with the target variable is dropped as a redundant variable. In one embodiment, if a categorical variable is highly correlated with any continuous one, the categorical variable is discarded. In this embodiment, the categorical variables are dropped rather than continuous variables because categorical variables are expanded into multiple dummy variables, which require greater processing time and system resources when building the statistical model.

[0018] In a further embodiment, when building a model, principle components are created and used instead of directly using the variables. As known in the art, principle components are linear combinations of variables and possess two main properties: (1) all

components are orthogonal to each other, which means no co-linearities exist among the components; and (2) components are sorted by how much variance of the data set they capture. Therefore, only important components (e.g., those exhibiting a significant level of variance) can be used to create a model. Empirical experiments show that including components, which represent 90% of the variance of a given data set, provides a sufficiently robust and accurate data model. In one embodiment, the number of these components to be included in creating the model can be less then n x 0.9 (where n is the number of all principle components). In this way, the size of the design matrix and processing time to build the model can be reduced. In a further embodiment, after the model is built based on the selected principle components, the coefficients of principle components are mapped back to the original variables of the data set to facilitate model interpretation and model deployment.

[0019] Thus, the method and system of the invention provides the ability to automatically analyze and process large data sets and create statistical models with minimal human intervention. As a result, users with minimal statistical training can build and deploy successful models with unprecedented ease.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Figure 1 illustrates a data matrix diagram representative of a data set containing a plurality of predictive variables and a target variable.

[0021] Figure 2 illustrates a flow chart diagram that provides a general overview of a process for automatically analyzing data and building a statistical data model, in accordance with one embodiment of the invention.

[0022] Figure 3 illustrates a flow chart diagram of a process of identifying and flagging categorical variables, in accordance with one embodiment of the invention.

[0023] Figure 4 illustrates a flow chart diagram of a process of performing automatic data analysis for model building, in accordance with one embodiment of the invention.

[0024] Figure 5 illustrates a flow chart diagram of a process of determining a best model construction path, in accordance with one embodiment of the invention.

[0025] Figure 6 illustrates a flow chart diagram of pre-processing continuous variables in a deployment data set, in accordance with one embodiment of the invention.

[0026] Figure 7 illustrates a flow chart diagram of pre-processing categorical variables in a deployment data set, in accordance with one embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0027] The invention is described in detail below with reference to the figures wherein like elements are referenced with like numerals throughout.

[0028] As used herein, the term "data" or "data set" refers to information comprising a group of observations or records of one or more variables, parameters or predictors (collectively and interchangeably referred to herein as "variables"), wherein each variable has a plurality of entries or values. A "target variable" refers to a variable having a range or a plurality of possible outcomes, values or solutions for a given problem or query of interest. For example, as shown in Figure 1, a data set 10 may include the following seven exemplary predictor variables: grades people received in their sixth grade math course (M); grades in sixth grade English (E); grades in sixth grade physical education (PE); grades in sixth grade history (H); grades in any elementary school art course (A); gender (G); and intelligence quotient (IQ). If information is gathered for a training set of m people, where "m" is a positive integer, then each of m observations has 7 entries, one for each variable, for a total of mx7 entries contained in the data set 10.

[0029] Figure 1 also illustrates an exemplary target variable (Y), which contains a plurality of ranges or range "bins" for a person's yearly income in U.S. dollars. In this example, the target variable may also be represented as a pure continuous variable having a range of $0 to some predetermined maximum value. It is possible that some of the predictor variables (M, E, PE, H, A, G, IQ) of the data set 10 have strong predictive value of the target variable (Y), while others have low predictive value. As well known in the art, the data set 10, or at least a subset thereof, can be used as "training data" to create a statistical model that provides a predictive correlation between the predictive variables and the target variable. It is understood that the variables illustrated in Figure 1 are exemplary only and that the invention is not limited to any particular type of variables or data. Rather, the invention may be utilized to automatically analyze any type of data or variables to determine whether they have statistical relevance to solving a particular problem or query.

[0030] In one embodiment, the steps required by a user to build a statistical model are minimized. The user simply connects to an ODBC-compliant database (e.g., Oracle, SQL Server, DB2, Access) or a flat text file and selects a data set or table for analysis. The user then specifies a field or name that serves as the unique identifier for the data set and a variable that is the target for modeling. This target variable is the variable of interest that is hypothesized to depend in some fashion on other fields in the data set. As another example, a marketing manager might have a database of customer attributes and a "yes" or "no" variable that indicates whether an individual has made purchases using the company's web portal. The marketing manager can select this "yes" or "no" field as the target variable.

[0031] Based on this data set and the target variable selected by the manager, the method and system of the invention can automatically build a model attempting to explain how the propensity to make online purchases depends on other known customer attributes. Some of the processes performed during this automatic model building process are described in further detail below, in accordance with various embodiments of the invention.

[0032] Figure 2 illustrates a flow chart diagram that provides a general overview of a method of automatically building a statistical model, in accordance with one embodiment of the invention. The process 100 begins at step 102 where training data comprising data set variables and at least one target variable are accessed or retrieved from memory of a computer system (not shown). As apparent to those skilled in the art, the methods and systems described herein are computer-based methods and systems and any computer or data processing system known in the art having sufficient processing power and capacity may be utilized to execute software that performs the steps and processes described herein.

[0033] In the art of statistical analysis, two common types of variables are "categorical" and "continuous" variables. The characteristics and differences between these two types of variables are well known in the art. At step 104, variables in the training set are analyzed and identified as either continuous or categorical variables and all categorical variables are flagged. Since categorical and continuous variables are typically treated differently when performing statistical analysis, in one embodiment, the user is given the opportunity to manually specify which variables in the data set are categorical variables, with all others deemed to be continuous. Alternatively, if the user is not a trained analyst or does not want to perform this task manually, the user can request automatic identification and flagging of "likely" categorical variables. In a

further embodiment, the user is given the opportunity to over-ride any categorical flags that were automatically created.

[0034] Next, at step 106, missing and outlier data is detected and processed accordingly, as described in further detail below. At step 108, exploratory analysis of the data is performed. At step 110, automatic analysis of the data to build a statistical model is performed. In one embodiment, this step is performed by an Automatic Model Building (AMB) algorithm or module, which is described in further detail below. At step 112, the results of the analysis of step 110 are then used by a core engine software module to build the statistical model. Next, at step 114, coefficients calculated during step 112 are mapped back to the original variables of the data set. Lastly, at step 116, the model is tested and, thereafter, deployed. Each of the above steps is described in further detail below.

[0035] In one embodiment, the steps of automatically analyzing a data set and building a model are performed in accordance with an Automatic Model Building (AMB) algorithm. Exemplary pseudo-code for this AMB software module is attached hereto as Appendix A. In preferred embodiments, the AMB software program provides the user with an easy to use graphic user interface (GUI) and an automatic model building solution.

[0036] As described above, the invention automatically performs various tasks in order to analyze and "cleanse" the data set for purposes of building a statistical model with minimal human intervention. These tasks are now described in further detail below.

<u>Identifying and Flagging Categorical Variables</u>

[0037] In one embodiment, a process for automatically identifying and flagging categorical variables (step 104) is performed in accordance with the exemplary pseudo code attached as Appendix B. In one embodiment, the field or record type (e.g., boolean, floating point, text, integer, etc.) is known in advance (e.g., data comes from a database with this information). Alternatively, if data comes from, e.g., a flat file, well-known techniques may be used to determine the types of fields or records in the data set.

[0038] In one embodiment, a solution to the problem of too many categories in a particular variable is to combine or "collapse" adjacent categories (e.g., A and B) when the max p-value for the adjacent categories is greater than or equal to Tmin, as provided for in the pseudo-code of Appendix B.

[0039] When a variable contains a large number of integer entries, it is often times difficult for an untrained user to determine whether it is a continuous or categorical variable. Figure 3 illustrates a flow chart diagram of a process of analyzing variables containing integer values in order to classify and flag such variables as either categorical or continuous, in accordance with one embodiment of the invention. At step 120, a variable i within a data set that contains integer values as entries is retrieved for processing. At step 122, the number of unique values within variable i with more than Nmin observations or entries (C(i, Nmin)) is calculated. Next, at step 124, it is determined whether C(i, Nmin) is less than or equal to a predetermined upper bound selected for a categorical variable (Cmax). If it is determined that C(i, Nmin) is less than or equal to Cmax, then at step 126, the variable i is flagged as a categorical variable and the process returns to step 120 where the next variable i containing integer values is retrieved for processing.

[0040] If at step 124, C(i, Nmin) is determined to be greater than Cmax, then at step 128 a query is made as to whether the variable i has significant predictive strength when treated as a continuous variable. Known techniques may be used to answer this query such as calculating the value of Pearson's r or Cramer's V for the variable with respect to the target variable. If it is determined that the variable i does have significant predictive strength when represented as a continuous variable, then at step 130, the variable is flagged as a continuous variable and the process returns to step 120 where the next variable i containing integer values is retrieved for processing until all such variables have been processed. Otherwise, the process proceeds to step 132 wherein a new variable i' is created by collapsing adjacent cells by applying a T-test criteria, in accordance with one embodiment of the invention.

[0041] At step 134, the process determines whether the number of unique values within the new variable i' (C(i', N)) is less than or equal to Cmax. If so, at step 136, variable i' is flagged as a categorical variable. Else, at step 138, the original variable i is flagged as a continuous variable. After either step 136 or 138, the process returns to step 120 where the next variable i containing integer values is retrieved for processing until all variables i having integer values as entries or observations have been processed.

[0042] As described above, if a user has limited information regarding the characteristics of potential predictor variables and/or if the user is untrained in the art of statistical data analysis, the invention provides a powerful tool to the user by automating the analysis and flagging of variable types for the user. Additionally, the invention safeguards against or minimizes the

potential of debilitating effects to model building that result when a user incorrectly or unwisely specifies a variable with hundreds of unique values as a categorical variable.

Handling Missing Data and Outliers

[0043] It is a rare and fortuitous occasion when a data set exhibits no missing values. More often missing values are encountered for many if not all the fields or variables in a data set. Some fields may have only a few missing values, while for others more than half of the values may be missing. In one embodiment, the method of the invention deals with missing values in one of two ways, depending on whether the field is continuous or categorical. For continuous variables, the method substitutes for the missing values the mean value computed from the non-missing entries and reports the number of substitutions for each field. For categorical variables, the invention creates a new category that effectively labels the cases as "missing." In many applications, the fact that certain information is missing can be used profitably in model building and the invention can exploit this information. In one embodiment, in the case of incomplete datasets, the missing counts of the severely missing observations are presented to the user (perhaps in a rank order format). She or he then has the option to either eliminate those observations or variables from the design matrix or substitute corresponding mean values in their place.

[0044] Outliers, on the other hand, are recorded data so different from the rest that they can skew the results of calculations. An example might be a monthly income value of $1 million. It is plausible that this data point simply reflects a very large but accurate and valid monthly income value. On the other hand, it is possible that the recorded value is false, misreported or otherwise errant. In most situations, however, it is impossible to ascertain with certainty the correct explanation for the suspect data value. In practice, a human analyst must decide during exploratory data analysis whether to include, exclude, or replace each outlier with a more typical value for that variable. In one embodiment, the invention automatically searches for and reports potential outliers to the user. Once detected, the user is provided with three options for handling outliers. The first option involves replacing the outlier value with a "more reasonable" value. The replacement value is the data value closest to a boundary of "reasonable values," defined in terms of standard deviations from the mean. Under the second option, the record (row) of the data set with the suspect value is ignored in building and estimating the model. The final option is to simply do nothing, i.e., leave data as is and proceed.

[0045] In one embodiment, outliers in a given (continuous) variable are identified by using a z-test with three standard deviations. For example, assume $x = (x_1, x_2, \ldots, x_M)$ is an observation vector for a variable and x_{mean}, sd are its mean and standard deviation, respectively, an entry x_i is considered an outlier if the following relation holds:

$$| x_i - x_{mean} | > 3 * sd$$

[0046] If a continuous variable has an exponential distribution, it should be log-scaled first before the outlier test above is conducted. The following pseudo-code describes the process:

```
For (each continuous predictor)
    If (the predictor is exponentially distributed)
        Log-scaling the predictor
    End If
    Perform the outlier detection
End For
```

[0047] As discussed above, in one embodiment, three handling options for outliers are provided to the user:

1. Substitute with MAX/MIN non-outlier value.
2. Keep (this is a do-nothing option)
3. Delete the corresponding record

[0048] In one embodiment, option 1 above is automatically selected as the default option and, during the training stage, outliers, once detected, are replaced by the highest (or lowest) non-outlier value within the vector, unless either of options 2 or 3 are manually selected by the user. In a further embodiment, in the default mode, these highest and lowest non-outlier values are also used to substitute any possible outliers (i.e., those outside the variable range in the training set) in the testing and deployment datasets as well. In a further embodiment, during deployment, all outliers are counted and a warning with an outlier summary is issued to the user.

Exploratory Data Analysis

[0049] Exploratory data analysis is the process of examining features of a dataset prior to model building. Since many datasets are large, most analysts focus on a few numbers called "descriptive statistics" that attempt to summarize the features of data. These features of data include central tendency (what is the average of the data?), degree of dispersion (how spread out is the data?), skewness (are a lot of the data points bunched to one side of the mean?), etc.

Examining descriptive statistics is typically an important first step in applied modeling exercises.

[0050] Univariate statistics pertain to a single variable. An exception is the sample correlation, which measures the degree of linear association between a pair of variables. Univariate statistics include well known calculations such as the mean, median, mode, quartiles, variance, standard deviation, and skewness. All of these statistical measures are standard and well-known formulas may be used to calculate their values. The correlation measure depends upon the underlying type of variables (i.e., it differs for a continuous-continuous pair and for a continuous-binary pair). Exemplary pseudo-code for computing common univariate statistical measures is provided in Appendix C attached hereto.

Identifying and Log-scaling Exponential Variables

[0051] In one embodiment, the models constructed by the invention are linear in their parameters. Linear models are quite flexible since variables may often be transformed or constructed so that a linear model is correctly specified. During the exploratory data analysis phase of a modeling project, statisticians frequently encounter variables that might reasonably be assumed to have an exponential distribution (e.g., monthly household income). Statisticians will often handle this situation by transforming the variable to a logarithmic scale prior to model building. In one embodiment, the method and system of the invention replicates this exploratory data analysis by determining for each variable in the data set whether an exponential distribution is consistent with the sample data for that variable. If so, the variable is transformed to a logarithmic scale. The transformed variable is then used in all subsequent model-building steps.

[0052] In most cases, linear regression modeling assumes all continuous variables are normally distributed. But in practice some of the given continuous variables can be exponentially distributed. In such circumstances, the AMB module detects and log-scales such exponentially distributed variables.

[0053] It is assumed that the log-scaling transformation helps convert an exponentially distributed variable (once detected) into a normally distributed variable. As a distribution test for a given variable, the variable is first sorted and then a sample of size n is selected with an evenly indexing distance. Then, the variable is log-scaled and the KS-test is used to determine whether it has a normal distribution. In one embodiment, detecting exponential variables is

performed in accordance with the exemplary pseudo-code illustrated in Appendix D attached hereto.

[0054] If a continuous variable is exponentially distributed, it is log-scaled in order to transform its distribution to a normal one. The log-scale formula roots from the following distribution test

$$x = 1 - e^{\frac{x - x\_\min}{x\_\min - x\_mean}}$$

where x_mean and x_min is a sample mean value and the sample minimum value of predictor x, respectively. In one embodiment, the step of log-scaling an exponentially distributed continuous variable is performed in accordance with the exemplary pseudo-code provided in Appendix E.

Auto-Analysis Of Data To Build Model

[0055] Referring again to Figure 2, after exploratory data analysis is completed at step 108, the method of the invention proceeds to step 110, where further analysis of the data is performed in order to build a statistical model. Figure 4 provides a flow chart diagram illustrating in finer detail some of the steps that are automatically performed in step 110 of Figure 2, in accordance with one embodiment of the invention.

[0056] As shown in Figure 4, at step 152, a univariate analysis is performed on each of the variables in the data set in order to filter out variables that have low correlation with the target variable. In one embodiment, the invention is embodied in a software program executed by a computer (not shown) wherein the program performs a multitiered variable filtration/selection process. In a first stage, the program applies a filter based on univariate predictive ability. The filter application varies with the number of potential predictors considered. For a relatively small number of predictors, no variables are removed. For larger sets of predictors, a subset of predictors with the worst univariate predictive performance is discarded. The objective at this stage is simply to reduce the number of potential predictors to a manageable size for further investigation. In one embodiment, univariate analysis is performed in accordance with the exemplary pseudo-code illustrated in Appendix F attached hereto.

[0057] Variables that survive the first filtration stage are then standardized. The motivation behind standardization is to maximize computational efficiency in subsequent steps. One advantage of standardizing the predictors is that the resulting estimated coefficients are unit-less,

so that a rescaling of monthly income from *dollars* to *hundreds of dollars*, for example, has no effect on the estimated coefficient or its interpretation.

[0058] Next, at step 154, the program bins continuous variables so that they may be compared to each of the categorical variables to determine whether the information contained in the categorical variables appears largely redundant when considered along with the continuous variables. Those categorical predictors that appear redundant are discarded, while those that remain are expanded into a set of dummy variables, i.e., variables that take the value 1 (one) for a particular category of the variable and the value 0 (zero) for all other categories of the variable.

[0059] In order to compare categorical variables with continuous variables, however, the continuous variables must first be "binned" per step 154. Since there is no direct correlation measurement for a continuous variable and a categorical variable, each continuous variable is binned into a pseudo-categorical variable and, thereafter, Crammer's V is applied to measure the correlation between it and a real categorical variable. In one embodiment, continuous values are placed into bins based on the position they reside on a scale. First, the number of bins (n) is determined as a function of the length of the vector. Then, the range of the continuous variable is determined and divided into n intervals. Lastly, each value in the continuous variable is placed into its bin according to the value range it falls in. In this way, the program creates an ordinal variable from a continuous one. If categories of a categorical variable are highly associated with the newly created ordinal variable, the Crammer's V will be high, and vice versa. In one embodiment, a continuous variable is binned in accordance with the exemplary pseudo-code provided in Appendix G attached hereto.

[0060] As discussed above, in one embodiment, during univariate analysis, some variables that are weakly related with target variable are filtered out of the data set. If there are both continuous and categorical variables, before merging them, the method of the invention attempts to eliminate the co-linearity between categorical variables and continuous ones.

[0061] After binning a continuous variable as discussed above, Cramer's V is used to evaluate the correlation between the binned continuous variable and a "real" categorical variable. In one embodiment, if the Crammer's V value is above a threshold, the categorical variable is discarded because categorical variables will typically be expanded into multiple dummies and occupy much more space in the design matrix than continuous variables. This process of eliminating categorical variables that are highly correlated with continuous variables is performed at step 156 in Figure 4. In one embodiment, the process of eliminating collinear

categorical variables is performed in accordance with the exemplary pseudo-code illustrated in Appendix H.

[0062] Next, after redundant categorical variables have been discarded in step 156, the program proceeds to step 158 and executes a process of expanding each of the remaining categorical variables into multiple dummy variables for subsequent model-building operations. One objective of this process is to assign to categorical variables some levels in order to take account of the fact that the various categories in a variable may have separate deterministic effects on the model response.

[0063] Typically there are multiple categories present in a categorical variable. Therefore, in one embodiment, the method of the invention assigns a dummy variable to each category (including the "missing" category) in order to build a linear regression model. A simple categorical expansion may introduce a perfect co-linearity. In fact, if a categorical variable has k categories and we assign k dummies to it, then any dummy will be a linear combination of the remaining k-1 dummies. To avoid this potential problem, in one embodiment, one dummy that is the *least* represented (in population), including a "missing" dummy, is eliminated. In one embodiment, the step of expanding categorical variables is performed in accordance with the exemplary pseudo-code provided in Appendix I attached hereto.

[0064] Next, at step 160, all continuous variables and dummies are normalized before further processing and analysis of the data is performed. To obtain principle components, the data set must first be normalized. After normalization, each variable becomes a unit-norm 1 and the sum of all entries of the variable is 0. For each variable x, in a vector format, the formula of normalization is as follow:

$x = \dfrac{x - \bar{x}}{\|x - \bar{x}\|}$, $\bar{x}$ is mean of x and $\|x\|$ is norm of x.

$\bar{x} = \dfrac{\sum_{i=1}^{n} x(i)}{n}$, n is the length of vector x

$\|x\| = \sqrt{\sum_{i=1}^{n} x(i)^2}$, n is the length of vector x

[0065] In one embodiment, the step of normalization is performed in accordance with the exemplary pseudo-code provided in Appendix J attached hereto.

[0066] At step 162, a second stage filtration of potential predictors involves examining the sample correlation matrix for the normalized predictors to mitigate the potential of multicollinearity and drop variables that are highly correlated with other variables. At this stage, the remaining variables are either continuous or dummy variables. Perfect multicollinearity occurs when one predictor is an exact linear function of one or more of the other predictors. The term multicollinearity generally refers to cases where one predictor is nearly an exact linear function of one or more of the other predictors. Multicollinearity results in large uncertainties regarding the relationship between predictors and the target variable, (large standard errors in null hypothesis test, which examines whether the coefficient on a particular variable is zero). In the extreme case, perfect multicollinearity results in non-unique coefficient estimates. In short, the second stage filter attempts to mitigate problems arising from multicollinearity.

[0067] In one embodiment, if two variables exhibit a high pairwise correlation estimate, one of the two variables is dropped. The choice of which of the pair is dropped is governed by univariate correlation with the target. While this procedure detects obvious cases of multicollinearity, it cannot uncover all possible cases of multicollinearity. For example, a predictor may not be highly correlated with any other single predictor, but might be highly correlated with some linear combination of a number of other predictors. By limiting consideration to pairwise correlation, models can be built more quickly, since searching for arbitrary forms of multicollinearity is often time consuming in large data sets. In other embodiments, however, when the time required to build a model is less of a priority, more comprehensive searching of multicollinearities may be performed to eliminate further redundant predictors or variables and build a more efficient and robust model.

[0068] Thus, in preferred embodiments, the method of the invention performs a second-stage variable filtering process after an initial variable screening has been performed. Some highly correlated variables (continuous and/or newly expanded dummies) are eliminated through the formulation of their normal equation matrix. Given a set of variables (or a design matrix), its normal equation represents a correlation matrix among these variables. For each pair of variables, if their correlation is greater than a threshold (in one embodiment, the default value is 0.8), then the pair of variables is considered to multicollinear and one of them should be eliminated. In order to determine which variable should be eliminated, the correlation values between each of the two predictive variables and the target variable are calculated. The predictive variable with a higher correlation to target value is kept and the other one is dropped.

In one embodiment, the process of eliminating multicolinearities is performed in accordance with the pseudo-code provided in Appendix K attached hereto.

[0069] At step 164, the normalized variables that survive the preceding filtration steps are combined into a data matrix and then Principle Components Analysis (PCA) is performed on this matrix. PCA techniques are well known in the art. Essentially, PCA derives new variables that are optimal linear combinations of the original variables. PCA is an orthogonal decomposition of the original data matrix, yielding orthogonal "component" vectors and the fraction of the variance in the data matrix represented or explained by each component.

[0070] In one embodiment, the invention applies a final filter by dropping components that account for only a small portion of the overall variance in the sample data matrix. All other components are retained and used to estimate and build a deployable model.

[0071] Since principle components are linear combinations of variables, the regression on components has several advantages over the direct regression on variables. First, all components should be orthogonal to each other and hence there is no co-linearity. This property helps build a more robust regression model. Secondly, since a portion of components can represent the most variance of a dataset, a relatively small component design matrix can be used for model building.

[0072] Given a normalized data set $X(m \times n)$, where $NE = X^T*X$ is its normal equation matrix, the following computation is performed:

$$[U\ S\ V] = SVD(NE);$$

where, $U(n \times n)$ is the loading matrix, each column is a singular vector of NE and V=U in this case; $S(n \times n)$ is a diagonal matrix that contains all singular values. The sum of the singular values is n. Next, a portion of the leading component from U is selected and $W = X*U$ is computed. The vector W is then used to build a regression model. In one embodiment, PCA processing is performed in accordance with the exemplary pseudo-code provided in Appendix L attached hereto.

Model Building Based On The Resulting Design Matrix

[0073] After PCA processing is completed, the invention is ready to build a model using the retained components in the data set. Referring again to Figure 2, at step 112, a core engine is executed to build the model.

[0074]

[0075] In one embodiment, the core engine utilizes the conjugate gradient descent (CGD) method and a singular value decomposition (SVD) method to generate a least squares solution. Both the CGD and SVD methods are model optimization algorithms that are well known in the art.

[0076] In one embodiment, the core engine has a two-layer architecture. In this architecture, a SVD algorithm serves as the upper layer of the engine that is designed to deliver a direct solution to the general least squares problems, while the CGD algorithm is applied to a *residual sum of squares* function and used as the lower layer of the engine. In one embodiment, the initial solution for CGD is generated randomly. This two-layer architecture utilizes known advantages of both the SVD and CGD methods. While SVD provides a more direct and quicker result for smaller data sets, it can sometimes fail to provide a solution depending on the quality or characteristics of the data. SVD can be slower than CGD for larger data sets. The CGD method, on the other, while requiring more processing time to converge, is more robust and in many cases will provide a reasonable solution vector.

[0077] In one embodiment, the upper-layer of the engine – an SVD approach for solving general least squares problems, is performed in accordance with the exemplary pseudo-code provided in Appendix M attached hereto.

[0078] Figure 5 illustrates a block diagram of the general decisions and processes performed by the core engine in accordance with one embodiment of the invention. At step 200, the engine determines whether the number of records in the data set is greater than 50,000. If yes, then at step 202, a SVD solution is computed to provide a direct solution to the general least squares problems. At step 204, the engine determines if the SVD computation was successful. If yes, the model building has successfully completed and the engine terminates at 210.

[0079] If it is determined that the number of records is greater than 50,000 (step 200) or the SVD computation was unsuccessful (step 204), then the engine utilizes the CGD method and, at step 208, calculates a random initial guess for a possible solution vector of the model. Next, at step 210, the CGD algorithm utilizes the initial random guess and applies its iterative algorithm to the residual sum of squares for the estimated target value and the observed target values.

[0080] The *residual sum of squares* is a function that measures variability in the observed outcome values about the regression-fitted values. The residual sum of squares is computed as

follows: assume that Y_j, $j = 1, 2, \ldots, K$ are the observed target values, and $\hat{Y}_j$, $j = 1, 2, \ldots, K$ the corresponding estimated values. Then, the residual sum of squares (in *func*) is defined as

$$\text{Func(X)} = \sum_{j=1}^{K}(\hat{Y}_j - Y_j)^2$$

[0081] In a further embodiment, the multidimensional derivative of the objective function is also used during CGD processing by the core engine. Both the function and the corresponding derivative are repeatedly used in CGD to iteratively determine a best possible solution vector for the model.

[0082] In one embodiment, the functional derivative of the residual sum of squares is calculated as follows: assume that X_{ij}, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, K$ is the value of the *jth* variable in the *ith* observation. Then, the functional derivative of the residual sum of squares (in *dfunc*) is given by

$$\text{dfunc}_j\text{(X)} = 2\sum_{i=1}^{M}(\hat{Y}_i - Y_i)X_{ij}, j = 1,2,\ldots,K$$

The procedure *dfunc* may take a solution vector as its input parameter and compute, through the design matrix, and return the multidimensional function derivative vector.

[0083] Referring again to Figure 2, at step 114, after a regression model has been built based on principle components, the PCA coefficients are "mapped back" to the original space of variables through the inverse of the loading matrix for the components before testing and deployment of the model. In one embodiment, component regression involves the following steps:

1. Normalized data set X(m x n), n variables selected from step 6 of the AMB algorithm (App. A);
2. NE = X^T*X is its normal equation matrix;
3. [U S V] = SVD(NE); We select some columns(say $k < n$ columns) of U in step 7 of AMB algorithm;
4. W = X*U; We used W(m x k) to build model and get the model coefficients, β_0, $\beta_1 \ldots \beta_k$;
5. y = f(β_0 + W*β) = f(β_0 + X*(U*β));

6. $U^*\beta$ is a vector of n by 1. Together with β_0 it forms the coefficient vector on variables, which will be presented to the user.

[0084] See Appendix A for further details. In one embodiment, the function of mapping coefficients back to the original variables is performed in accordance with the following exemplary pseudo-code:

Input: 1. Model coefficient on Principle Components $\bar{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$

2. Loading matrix $U(n \times k)$;

Output: 1. Model coefficient on variables $\bar{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n)^T$

Parameter : None

Process:

$\alpha 0 = \beta_0$;

$(\alpha_1, \ldots, \alpha_n)^T = U * (\beta_0, \beta_1, \ldots, \beta_k)^T$

return $\bar{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n)^T$

Model Testing and Deployment

[0085] Now that the model is built, it is ready for step 116 where the model is tested (validated) and deployed. A first task is to pre-process the test/deployment dataset to a format and structure which is the same as that of the training set. If the test data set is a subset of the original data set for which pre-processing has already been performed then the following steps may be omitted for executing the model on the test data.

[0086] If a record in the deployment set has missing values, or contains values outside the ranges defined by the training set, it will be marked invalid, and a summarized report is issued to the user. Before applying a model to a dataset, the data must be pre-processed and formatted in the same way as the training data set. Based on the variable attributes and information collected during the exploratory data processing on the training set, the invention preprocesses and then scores a deployment dataset. For example, if an original raw variable is not selected during model building, it will be dropped and not processed during deployment.

[0087] Figure 6 illustrates a flow chart diagram for preprocessing continuous variables, in accordance one embodiment of the invention. From starting point 300 the process proceeds to step 302 to determine whether a current variable was selected (i.e., survived filtration/elimination) during the model building process. If no, at step 304, the variable is

dropped and not considered further. If the variable was a selected variable, at step 306, the process queries whether it is a "missing" variable. If no, then at step 308, outliers are detected and handled. Next, at step 310, the process queries whether the variable has an exponential distribution and needs to be log-scaled. If no, then at step 312, the mean value and normx value is retrieved to normalize the variable. At step 314, the variable is normalized and, at step 316, the process obtains the design matrix column location or index for the variable and puts the variable in a corresponding column in a deployment data matrix. Thereafter, the process is done.

[0088] If at step 306, the variable is determined to be a missing value, then, at step 318, the process retrieves the saved mean value of the variable calculated from the training set. Next, at step 320, the missing value is substituted with the mean value and the process moves to step 310. If, at step 310, it is determined that the variable is exponentially distributed and requires log-scaling, then, at step 322, the process retrieves a saved mean value of a predetermined number of samples of the variable from the training set as well as a minimum value of samples . Then, at step 324, these values are used to log-scale the variable. The process then performs steps 312-316 as described above and, thereafter, is done.

[0089] Figure 7 illustrates a flow chart diagram for a method of pre-processing categorical variables for deployment, in accordance with one embodiment of the invention. The process starts at 400 and, at step 402, queries whether any dummy variables have been retained in the training set for that variable during model building. If no, then, at step 404, the variable is dropped from further consideration. If dummy variables were retained during model, then at step 406, the process retrieves the column index range for the variable. Next, at step 408, the columns in this range are initialized with "0's." At step 410, the process queries whether the current dummy variable appears in the training set. If a particular dummy does not appear in the training set, then at step 412, the process queries whether the column index for that dummy is greater than 0. If yes, then at step 416, a "1" is assigned in the corresponding entry in the design matrix. If the answer is no, then at step 418, the process retrieves the save mean value and normx values for normalization. Then, at step 420, the process normalizes all 1's and 0's in the range, x=(x-mean)/normx.

[0090] If at step 410, it is determined that the dummy did appear in the training set, then at step 414, the process retrieves the column index of the dummy variable in the training data matrix or a data design matrix. Note that in one embodiment, the training data matrix is a subset

of the data design matrix, which also contains test data that is subsequently used for testing the model. After step 414 is completed, the process then proceeds to step 412 and executes steps 412 and 416-420 as described above.

[0091] In one embodiment, a method of pre-processing continuous and categorical variables, respectively, for deployment is performed in accordance with the exemplary pseudo-code provided in Appendix N attached hereto.

Model Output Statistics

[0092] When many potential predictors are used in building a model, there is always the potential for over-fitting. Over-fitting refers to fitting the noise in a particular sample of data. The concern of over-fitting is that in-sample explanatory power may be a biased measure of true forecasting performance. Models that over-fit will not *generalize* well when they make predictions based on new data. One remedy for the problem of over-fitting is to split the data set into two subsets prior to estimating any unknown model, one dubbed the "training" set and the other the "validation" set. Model parameters are then estimated using only the data in the training subset. Using these parameter estimates, the model is then deployed against the validation set. Since the validation data are effectively new, model performance on this validation set should provide a more accurate measure of how the model will perform in actual practice.

After the model has been built and tested, model output statistics can be computed in order to provide a set of useful summary measures in describing, interpreting and judging the resulting model. In one embodiment, these output statistics are classified into two categories: one is associated with individual model coefficients; the other is related to the overall regression model (as an entity).

[0093] In one embodiment, most of the standard and important statistics for general linear regression models that typically can be found in popular statistics software packages are outputted. The following lists these model output values with some brief descriptions.

Category I (Model Coefficient)

- Predictor names
- Standard Error (SE) for each model coefficient – The square root of variance for each estimated regression coefficient. SE is computed only on the training set.

- Estimated model coefficients – the model solution vector
- Confidence interval for each model coefficient – An interval estimation that provides a range of possible values with a certain level of confidence. CI is computed only on the training set.
- Significance t-test for each model coefficient (H_0: coefficient = 0) – a hypothesis test on individual regression coefficient. T-test as well as its p-values are computed only on the training set.

Category II (Overall Model)

- R^2 – (including SSE and SSR) this is the most popular performance metric for linear regression models. It measures the proportion of total variation about mean of target values explained by the regression. Another name for R^2 is the *coefficient of multiple determination.* Note that $0 <= R^2 <= 1$. The larger it is, the better the fitted regression equation explains the variation in the data. R^2 is computed on both sets (testing and training).
- Adjusted R^2 – A related statistics, which might be more suitable in some applications, is the *adjusted* R^2. It is the R^2 weighted by the number of independent variables and observations.
- AIC (Akaike information Criterion) – It is a model performance metric that can be used to compare different fitted models. The smaller the AIC, the better the fit. AIC also provides a balance between the fit and the number of predictors.
- As a final model fitness metric, AIC is computed on the testing set. However, AIC is repeatedly computed on the training set for variable selection inside the stepwise process.
- BIC (Bayesian Information Criterion, also known as Schwarz's Bayesian Criterion (SBC)) – It an alternative to AIC. As a final model fitness metric, BIC is computed on the testing set. However, BIC is repeatedly computed on the training set for variable selection inside the stepwise process.
- Significance F-test - It is a hypothesis test on the overall regression equation (the hypothesis that all regression coefficients are significant at some confidence level). F-test as well as its p-values are computed only on the training set.
- Mean Squared Error (MSE) – a statistic used to measure the efficacy of prediction. MSE is computed on both sets (testing and training)

- Sum of Squares of the Error (residual) (SSE). SSE is computed only on the testing set.
- Sum of Squares due to the Regression (SSR). SSR is computed only on the testing set.
- A further detailed discussion about computing performance statistics for a model based on a data matrix X is provided below. Assume X *is* a given design matrix (including the bias term) of dimension M by N, $y= (y_1, y_2, \ldots, y_M)$ are the observed target values, $y_{mean} = mean(y)$, and $z = (z_1, z_2, \ldots, z_M)$ the predicted target values.

SE for each model parameter

[0094] The estimated coefficient $\hat{\beta}_j$ is normally distributed with variance $\sigma^2 v_{jj}$, where v_{jj} is jth diagonal entry of $(X^T X)^{-1}$. σ^2 is unknown and its estimate is given by

$$s^2 = \frac{\sum_{i=1}^{M} (z_i - y_i)^2}{M - N}$$

Then,

$$SE_j = s\sqrt{v_{jj}}, \quad j = 1, 2, \ldots, N$$

Confidence Interval (CI) for each model parameter

[0095] A 100(1-α) % CI ($\alpha = 0.05$) on the coefficient β_j is given by

$$\hat{\beta}_j \pm t_{M-N,\alpha/2} SE_j, \quad j = 1, 2, \ldots, N$$

where $t_{M-N,\alpha/2}$ is the upper $\alpha/2$ critical point of the t-distribution with M-N d.f. and SE_j is the estimated standard error for the coefficient.

Significance t-test

[0096] Under the hypothesis that the coefficient β_j is zero, an α-level t-test rejects the hypothesis if

$$|t_j| = \frac{|\hat{\beta}_j|}{SE_j} > t_{M-N,\alpha/2}, j = 1,2,\ldots,N$$

where $t_{M-N,\alpha/2}$ is the upper $\alpha/2$ critical point of the t-distribution with M-N d.f. and SE_j is the estimated standard error for β_j.

R²

[0097] It can be defined as

$$R^2 = 1 - \frac{SSE}{\sum_{i=1}^{M}(y_i - y_{mean})^2}$$

SSE (Sum of Squares of the Error (residual))

$$\text{SSE} = \sum_{i=1}^{M}(z_i - y_i)^2$$

SSR (Sum of Squares due to the Regression)

$$\text{SSR} = \sum_{i=1}^{M}(z_i - y_{mean})^2$$

[0098] When over-fitting, the above R^2 can be a negative value and in this case it should be reset to zero. The R^2 measure is applied to both the training set and the testing set. If there is a large discrepancy in R^2 between these two sets, which likely indicates that an over-fitting model is generated, the system will issue a model-over-fitting warning to the user.

Adjusted R²

[0099] It is given by

$$R_a^2 = 1 - (1 - R^2)(\frac{M-1}{M-N}),$$

AIC

[0100] It is given by

$$\text{AIC} = \left(\log 2\pi + \log \frac{\sum_{i=1}^{M}(z_i - y_i)^2}{M} + 1 \right) + \frac{2N}{M}$$

BIC

[0101] It is given by

$$BIC = \left(\log 2\pi + \log \frac{\sum_{i=1}^{M} (z_i - y_i)^2}{M} + 1 \right) + \frac{\log(M)N}{M}$$

Significance F-test

[0102] The F-statistic can be defined as

$$F = \frac{\sum_{i=1}^{M} (z_i - y_{mean})^2 /(N-1)}{\sum_{i=1}^{M} (z_i - y_i)^2 /(M-N)}$$

[0103] It can be shown that when the hypothesis (that all regression coefficients are insignificant) is true, the F statistic follows an F-distribution with N-1 and M-N d.f. Therefore an α-level test rejects the hypothesis if ($\alpha = 0.05$)

$$F > f_{N-1, M-N, \alpha}$$

where $f_{N-1, M-N, \alpha}$ is the upper α critical point of the F-distribution with N-1 and M-N d.f..

MSE

[0104] It is defined as

$$MSE = \frac{\sum_{i=1}^{M} (z_i - y_i)^2}{M}$$

[0105] Various embodiments of a new and improved method and system for performing statistical analysis and building statistical models are described herein. However, those of ordinary skill in the art will appreciate that the above descriptions of the preferred embodiments are exemplary only and that the invention may be practiced with modifications or variations of the devices and techniques disclosed above. Those of ordinary skill in the art will know, or be able to ascertain using no more than routine experimentation, many equivalents to the specific embodiments of the invention described herein. Such modifications, variations and equivalents are contemplated to be within the spirit and scope of the present invention as set forth in the claims below.